QPack web services

QPack Web services user manual

ORCANOS
Application Lifecycle Management

1 Overview....4
2 Web service list....4
2.1 About....4
2.2 Add_Attachment....5
2.3 Add_Defect....7
2.4 Add_Embeded_Image....10
2.5 Add_Object....11
2.6 Add_object_discussion....15
2.7 Add_Relation....15
2.8 Add_Step....17
2.9 Check_Login_User....18
2.10 Check_Object_Lock_And_Status....18
2.11 Check_Silent....18
2.12 Delete_Object....19
2.13 Get_Chart_Results....19
2.14 Get_Children....19
2.15 Get_Data....21
2.16 Get_Data_By_Paging....21
2.17 Get_Execution_List....22
2.18 Get_Execution_Run_Details....23
2.19 Get_Execution_Run_Details_xml....26
2.20 get_filter_list_xml....30
2.21 Get_Item_Details....31
2.22 Get_Item_Details_xml....31
2.23 Get_Object....31
2.24 Get_Object_Attachments....33
2.25 Get_Object_Audit_Trail....34
2.26 Get_Object_Definition....36
2.27 Get_object_discussion....37
2.28 Get_object_metadta....38
2.29 Get_Object_Relations....38
2.30 Get_Projects....39
2.31 Get_Qpack_Xml....41
2.32 Get_Source_Report....41

2.33 Get_Unit_Execution_List 41
2.34 Get_User_Permissions_ByItemType 42
2.35 Get_user_projects 42
2.36 Get_User_Projects_Web 42
2.37 Get_Users 42
2.38 Get_Validation 44
2.39 Get_WebServ_Ver 44
2.40 GetActivityListByProjectVersionID 44
2.41 GetALLObjTypes 45
2.42 getFTPSettings 45
2.43 getProjectVersionList 45
2.44 GetTree 45
2.45 Init 45
2.46 Insert 45
2.47 Insert_BYCmd 46
2.48 Lock_object 46
2.49 Record_Execution_Results 46
2.50 Send_Mail 48
2.51 Subscribe_Item 48
2.52 TestAccessConnection 48
2.53 TestConnection 49
2.54 TestInsert 49
2.55 TestUpdate 49
2.56 Unlock_object 49
2.57 Update 49
2.58 Update_Defect 49
2.59 Update_Object_By_ID 49
2.60 Update_Server_ConfigXML 52
2.61 UpdateCommitRevisionNumber 52

ORCANOS
Application Lifecycle Management

1 Overview

Overview

QPack API's are web services, developed in .Net 3.5 environment.
Web services files located in c:\program files\QPack\QPackServ

URL address: http://YourServer/qpack/qpackserv/qpackserv.asmx

Example for executing web service (Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl = "http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.add_Step ("pola", "1234",test_id,"", "description", "expected results")
```

2 Web service list

2.1 About

Purpose

The about screen will show the version of the WS by reading the information from the Server side

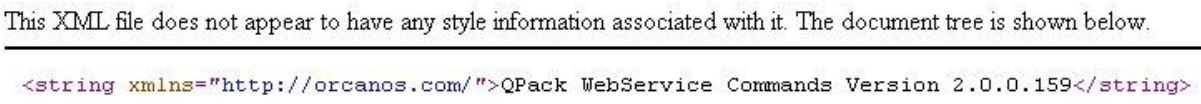
General Description

Input

None

Field name	Type	Mandatory	Description

Output



2.2 Add_Attachment

Purpose

Add attachment to object/defect

General Description

The web service will add attachment to object/defect
No branch is created upon adding attachment – add to all item versions

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Object_ID	Long	Yes	Item/defect ID
Object_Type	String	No	OBJECT/DEFECT. Default value is OBJECT Please note: Use capital letters.
Attachment_Name	String	Yes	File name or URL. File extension is mandatory Please note: URL is not supported

Field name	Type	Mandatory	Description
Attachment_Type	String	No	FILE/URL. Default value is File. Please note: URL is not supported
Attachment64	64 encode	No	Attachments in 64 encode. Mandatory if attachment type is File
Attachment_Description	String	No	Attachment description
Created_By	String	No	The user that add the attachment. Default is the input User_Name
Created_Date	String	No	Date format: **DD/MM/YYYY** Default value is Now
Over write	String	No	Yes/ No Default value is Yes

Output

If succeed:

```
<Attachment ID="14" Name="Filename.txt"/>
```

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
```

```
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res =
objSoap.Add_Attachment("pola","1234","17","","fileName.txt","","GYT
FHJ9n","Description","","")
```

Open issues

1.Adding attachments of File type only (Exclude URL attachments)
2.Not handling branching
3.Adding attachments to objects only (Without Defects) - **This issue was removed since ver 2.0.0.103**

2.3 Add_Defect

Purpose

Add new defect to QPack.

General Description

The webservice will save system fields like assign to, status, etc.
The webservice will save up to 15 custom fields
The webservice will make new entry in history table

Input

Field name	Type	Mandatory	Description
User_name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_name	String	Yes	Defect project name.
Major_Version	Long	Yes	The major version view number as appears in QPack.
Minor_Version	Long	Yes	The Minor version view number as appears in QPack.
Parent_ID	String	No	Input the parent item id where defect will be created under.

Field name	Type	Mandatory	Description
			Default value will be the input project
Synopsis	String	Yes	Defect synopsis
Description	String	No	Defect description in plain text/HTML format
Defect_type	String	No	Type of defect – "Internal"/"External". Default is "Internal" [External stands for defects reported by customers]
Assigned_to	String	No	User name the defect should be assigned to. User must belong to project. Default is the default assigned user in status work flow
Created_date	String	No	The defect created date. **Format must be the date format of the local machine** Default is Current date & time
Created_by	String	No	The name of the user created the defect. Default is the user that calls the service.
Category	String	No	Defect category. Default is category default value (Taken from admin system tables)
Severity	String	No	Defect severity. Default is severity default value (Taken from admin system tables)
Status	String	No	Defect status. Default is status default process flow value (Taken from admin system tables)
Priority	String	No	Defect priority. Default is priority default value (taken from admin system tables)
Found_version	String	No	Defect found version. Format: 99.99.999.9999

Field name	Type	Mandatory	Description
			Default is input version (major_version.minor_version.0.0)
Fixed_version	String	No	Defect fixed version. Format: 99.99.999.9999 Default is input version (major_version.minor_version.0.0)
Target_version	String	No	Defect target version. Format: 99.99.999.9999 Default is input version (major_version.minor_version.0.0)
Start_date	String	No	The defect start date. **Format must be the date format of the local machine** Default is Current date & time
Effort_estimation	Long	No	Defect effort estimation. Default is 0
Due_date	String	No	The defect due date. **Format must be the date format of the local machine** Default is Current date & time
Solution	String	No	Defect resolution
Work_around	String	No	Defect work around
CS1_Name	String	No	Defect 1st custom field name.
CS1_value	String	No	Defect 1st custom field value
Up to CS15			Up to 15 custom fields
MigrationRreference	String	No	For each item inserted, you can add a reference string. Later on you will be able to use this reference in reports, or in order to delete group of items - you will be able to delete group of items by the migration reference string
External_id	String	No	If the defect source is from 3rd party system. You can add the ID from the 3rd party system for later use.

Output

Defect ID or error message (In xml format)

Example(Visual Basic):

```
Dim objSoap asNew MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string

sUrl ="http://serverName/QPack/QPackserv/QPackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.add_defect ("pola", "1234", "ProjectName", 1, 0, , "Defect synopsis", "Defect description", "External", "pola", , , , , , , , , , , , , , , "Defect solution", "defect work around", "Custom field name 1", "custom field value 1")
```

Open issues

2.4 Add_Embeded_Image

Purpose

Add image located in item description to QPack FTP server
See Input/Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Add_Embeded_Image

General Description
The web service will return the path of the file to put in the description, in the followong format: ftp://servername/filename.

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password

Field name	Type	Mandatory	Description
Object_ID	Long	Yes	Selected item id
File_Name	String	Yes	file name and extension such as flowchart.gif
File_Content	String	Yes	64bit conversion of the file

Output

If succeed:

```
<File name="URL of the added file"/>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res =
objSoap.Add_Embeded_Image("pola","1234","33444","flow.gif","fsfsdf
sdfdsfdsrewrf4rwfrfwf")
```

2.5 Add_Object

Purpose

Add new object to QPack.
See Input/Output in http://[Server Name]/QPack/qpackserv/qpackserv.asmx?op=Add_Object

General Description

The web service will save system fields like assign to, status, etc.
The web service will save up to 15 custom fields
The web service will make new entry in history table

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_ID	Long	Yes	Project ID.
Major_Version	Long	Yes	The major version view number as appears in QPack.
Minor_Version	Long	Yes	The Minor version view number as appears in QPack.
Object_Name	String	Yes	The item name
Object_Type	String	Yes	Type of object as appear in the system (exclude defect): Requirement, test case, task, High level design, change request, unit test)
Release_Version	Long	No	Release value (major.minor.**RELEASE**.build) Default is 0
Build_Version	Long	No	Build value (major.minor.release.**BUILD**) Default is 0
Description	String	No	Defect description in plain text/HTML format
Parent_ID	Long	No	Input the parent item id where item will be created under. Default value will be the input project

Field name	Type	Mandatory	Description
Assigned_to	String	No	User name the object should be assigned to. Default is "Not Assigned"
Created_date	String	No	The object created date. **Format must be the date format of the local machine** Default is Current date & time
Created_by	String	No	The name of the user created the object. Default is the user set as project manager
Category	String	No	Specific object category. Default is category default value (Taken from admin)
Status	String	No	Valid object status. Default is status default process flow value (Taken from admin)
Priority	String	No	Valid object priority. Default is priority default value (Taken from admin)
Start_Date	String	No	The object start date. **Format must be the date format of the local machine** Default is Current date & time
Effort_Estimation	Long	No	Object effort estimation. Default is 0
Due_date	String	No	The object due date . **Format must be the date format of the local machine** Default is Current date & time

Field name	Type	Mandatory	Description
CS1_Name	String	No	Object 1st custom field name.
CS1_value	String	No	Object 1st custom field value
Up to CS15			Up to 15 custom fields
Insert_to_Pool	String	No	**For change requests/requirements only**: Determine whether item is inserted to pool (value=Y) of to product tree (value=N) Default: N
Migration_Reference	String	No	For each item inserted, you can add a reference string. Later on you will be able to use this reference in reports, or in order to delete group of items - you will be able to delete group of items by the migration reference string
External_ID	String	No	If the object source is from 3rd party system, you can add the ID from the 3rd party system for later use.

Output

If succeed:

```
<Key>REQ-39</Key>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/QPack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.add_object ("pola", "1234",2, 1, 0,"Object
name","Requirement")
```

Open issues

2.6 Add_object_discussion

This service adds discussion to an item.

Parameter	Value
sUserName:	QPack user name, mandatory (string)
sItemKey:	QPack item to which discussionis added, mandatory (string)
sComment:	

2.7 Add_Relation

Purpose

Add new relation between objects

General Description

The web service will add relation between objects
If relation is updated – change the timestamp of the object
Save to history
Only objects – doesn't include defects

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Source_ID	Long	Yes	The relation source item ID
Target_ID	Long	Yes	The relation target item ID
Relation_Type	String	No	Relation type. Default is relation type default value (Taken from admin)
Comment	String	No	Relation comment
Score	Long	No	Relation score. Default is 0

Output

If succeed:

```
<Relation status="OK"/>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Relation ("pola", "1234","21","23","","","")
```

Open issues

1.Not include checking relation constraint (Items status constraints)
2.Not include checking related items versions (In order to add relation to branched/original items)

2.8 Add_Step

Add Step API

Purpose: Add step to existing test or defect

Output: Step ID or error message (In xml format)

Input:

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Object_id	Long	Yes	Input the parent item id (test case/defect) where step will be created under.
Object_Type	String	No	Input the parent item type (test case/defect) where step will be created under. Default value is test case
Step_Description	String	Yes	Step description
Step_Expected_Result	String	Yes	Step expected results

Example (Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string

sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"

objSoap.MSSoapInit (sUrl & "?wsdl")

res = objSoap.add_Step (“pola”, “1234”,test_id,"", "description", "expected results")
```

2.9 Check_Login_User

Description:

This web service purpose is to check user login validation - User name , password , license

Input

Field name	Type	Mandatory	Description
sUserName	String	Yes	user name
sPassword	String	Yes	password
lProjectID	Int	Yes	project id
lSolutionVersion	Int	Yes	If enter to solution then solution version ID (Version_id column from PROJECT_VERSION table). If enter to project - 0
iApplicationType	Int	Yes	0 - User application 1 - MRD application 2 - Admin
sHostName	String	Yes	PC name

2.10 Check_Object_Lock_And_Status

This service check object status and lock user (QPack use).

Parameter	Value
Object_ID:	Object id to check, mandatory (int)
Version_ID:	Version id, mandatory (int)

2.11 Check_Silent

This service checks whether silent login is allowed for a selected user

2.12 Delete_Object

Purpose
Delete Object from QPack

General Description
The web service will delete object from QPack
Deletion include:
1.Delete object relation
2.Delete object attachments
3.Delete object steps (If has)
4.Remove object defects

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Object_ID	Long	Yes	The select object ID

Output

2.13 Get_Chart_Results

This service return a xml with the chart results (QPack use)

Input:
Record name as string (Record set saved name)
Chart type as int (0-Simple bar , 1-Multi bar , 2-Pie)
Column_1 as string - the grouped by column
Column_2 as string - the second grouped by column (Used only in multi bar chart)
Reference table 1 - column_1 reference table (In has one)
Reference table 2 - column_1 reference table (In has one)

Output:
Results - Xml string

2.14 Get_Children

Purpose

Return all children of selected item (one level)

General Description

The web service will return all object's children (first level only) with all children details

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_Name	String	Yes	Project name
Version	String	Yes	The version view of the objects. Format shall be : [Major version]. [Minor version] (2.0 for ex.)
Parent_ID	Long	Yes	Selected item id

Output

If succeed:

```
<Parent Id=”432432” ChildCount="0">
    <object id=”3232” action="">
        <field Name=”Object name” Value =”New object 1” />
        <field Name=”Description” Value =”Description of object 1” />
        <field Name=”Description64” Value =”43242343242342342342343242 3” />
        …
    </object>
    <object id=”12” action="Copy as link">
        <field Name=”Object name” Value =”New object 2” />
        <field Name=”Description” Value =”Description of object 2” />
        <field Name=”Description64” Value =”43242343242342342342343242 3” />
        …
    </object>
   …
</Parent>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

action = "Copy as link" - item copy as link and it's not the original item
action = "Head" - item is in pool (or risk)

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Children ("pola", "1234","Qpack","2.0",28)
```

2.15 Get_Data

This service return a record set results (QPack use)
Input: Query (String)
Output: Record set as xml

1.Open connection to DB
2.Get records by query
3.Sent the record set as xml

2.16 Get_Data_By_Paging

This service return a record set results by paging (QPack use)

This service save the record set in local folder in the server named: Cache_Results
Record set name is unique for every user and every grid and control

Input:
Query as string
Record name as string (Record set saved name)
Page number
Result per page
Refresh indication - if 1 the refresh the records. if 0 the get old data that saved in the server as file

Filter by (optional)- In order to get drill down from graph results
Sort by (optional) - In order to sort

Output:
Record count (Write as [Record count])
Record set - Xml string

2.17 Get_Execution_List

Purpose
This web service will return list of execution sets from specific project

General Description
This web service will return list of execution sets from specific project

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_Name	Long	Yes	Project Name
Version_View	String	Yes	The version of the selected project (2.0 for ex.)

Output

```
<execution_list>
    <execution Id=”1222” >
        <Name>Execution set 1</name>
        <Description>[CDATA]HTML description of execution[CDATA]/Description>
        <parentId>4433</parentId>
        <parentName>Requirement1<parentName>
    </execution>
    <execution Id=”3333” >
        <Name>Execution set 2</name>
        <Description>[CDATA]HTML description of execution[CDATA]/Description>
        <parentId>4444</parentId>
        <parentName>Requirement2<parentName>
    </execution>
</execution_list>
```

Example(Visual Basic):

Open issues

2.18 Get_Execution_Run_Details

Purpose

This web service will return full execution set content including test cases connected.

General Description

This web service will return all the details of the execution set to be run.

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Execution_Set_ID	Long	Yes	The execution to retrieve
Run_Status_Filter	String	No	Filter by list of statuses, separated by [,], for example: “No run, Fail”. Default is

Field name	Type	Mandatory	Description
			empty=all
Assigned_Filter	String	No	Filter by list of users, separated by [,], for example: "user1, user2". Default is empty=all
Work_Status_Filter	String	No	Filter by list of work status, separated by [,], for example: "sts1, sts2". Default is empty=all
Test_Id	String	No	List of test cases in the execution (multiple select, examples: "12121, 343, 555"). Default=All
TestInExecLineID	String	No	List of test instances to execute (multi select, for example: "3222, 4433, 555)
Type	String	No	"Auto"/"Manual", default="Auto"

Output

```
<Execution_Set_Run_Results id="4343" name="exec 1" cycle="1">
    <Run name = "12_03_2010_12_00_04" version="1.2.3.2" start="" end
="" user="user1" Type="Manual" TestCount ="4">
     <TestInExecLine ID  ="444" order = "1" >
            <Test id="44333"></Test>
            <ImportStatus>Will be filled by QPack when recording results</
ImportStatus >
            <ErrorDesc> Will be filled by QPack when recording results </ErrorDesc>
            <Name>Test name</Name>
            <LastRunStatus>Last run status</LastRunStatus>
            <Description>HTML</Description>
<steps>
              <step id="55444" Order="1" >
                  <Description>HTML</Description>
                  <Expected>HTML</Expected>
                  <actual>Will be filled by user</actual>
                  <Command>c:\a.bat</command>
                  <commandParams>p1 p2 p3</commandParams>
                  <SQLQry>select * from table</SQLQry>
                   <Run Status="No run"/>
                  <time start=" Will be filled by user" end=" Will be filled by user" />
               </step>
               <step id="55445" Order="2" >
                  <Description>HTML</Description>
                  <Expected>HTML</Expected>
                  <actual>Will be filled by user</actual>
                  <Command>c:\a.bat</command>
                  <commandParams>p1 p2 p3</commandParams>
                  <SQLQry>select * from table</SQLQry>
                   <Run Status="No run"/>
                  <time start=" Will be filled by user" end=" Will be filled by user" />
               </step>
            </steps>
            <InputTestParams>
               <param name="Band" value="2.5"/>
               <param name="BandWidth" value="35"/>
            </InputTestParams>
            <ResultParams >
               <param name="filled by user" Expected="by user" Actual="by user"/>
                More results Will be filled by user (include Run & test parameters)
            </ResultParams >
      </TestInExecLine >
```

Cycle is the one shown in the execution set (the last one)
Run name is a unique string from datetime parsing - created for each call for this web service and will be used to record the results with unique value.
Run version default is taken from the execution set version.
User is the login user to the Web Service (User_Name)
TestInExecLineID – the id of the line of the test in the execution. Each line is per test (can be duplicate due to parameters instances or due to double test in the execution).
ImportStatus – Success/Fail/Empty - Used when saving the results. Each test that was saved successfully is marked Success. Each test that was failed in save marked as Fail. Default is empty.
Test Params - Note that they will be added only if the instances of the test where added to the test before it added to the execution set (the test exist in the exe-set as instance). It will not be added if the test has parameters but not instances.
Results Params - Will be written after the XML is created and will be used for 'Record_execution_Results' in order to define and simulate **new** parameters. Not yet implemented

Example(Visual Basic):

Open issues
Notes:
- Project & Version is not relevant since each execution in different versions has unique Id. Even if not, the execution was get different Id when branched (or save changes to previous versions).
- When there is no test to write to the XML, the header is written anyway.
- Result Parameters

2.19 Get_Execution_Run_Details_xml

Purpose
This web service will return full execution set content including test cases connected.
This web service comes in addition to the web service 'Get Execution Run Details',
The difference between the 2:
User will get the results in plain XML and will have the ability to remove HTML tags

General Description

This web service will return all the details of the execution set to be run.

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Execution_Set_ID	Long	Yes	The execution to retrieve
Run_Status_Filter	String	No	Filter by list of statuses, separated by [,], for example: "No run, Fail". Default is empty=all
Assigned_Filter	String	No	Filter by list of users, separated by [,], for example: "user1, user2". Default is empty=all
Work_Status_Filter	String	No	Filter by list of work status, separated by [,], for example: "sts1, sts2". Default is empty=all
Test_Id	String	No	List of test cases in the execution (multiple select, examples: "12121, 343, 555"). Default=All
TestInExecLineID	String	No	List of test instances to execute (multi select, for example: "3222, 4433, 555)
Type	String	No	"Auto"/"Manual", default="Auto"
convertHTMLtoText	String	Yes	TRUE/FALSE Change the html tags to plain text use values 'true' or 'false'

Output

```
<Execution_Set_Run_Results id="4343" name="exec 1" cycle="1">
    <Run name = "12_03_2010_12_00_04" version="1.2.3.2" start="" end
="" user="user1" Type="Manual" TestCount ="4">
     <TestInExecLine ID  ="444" order = "1" >
            <Test id="44333"></Test>
            <ImportStatus>Will be filled by QPack when recording results</
ImportStatus >
            <ErrorDesc> Will be filled by QPack when recording results </ErrorDesc>
            <Name>Test name</Name>
            <LastRunStatus>Last run status</LastRunStatus>
            <Description>HTML</Description>
<steps>
              <step id="55444" Order="1" >
                  <Description>HTML</Description>
                  <Expected>HTML</Expected>
                  <actual>Will be filled by user</actual>
                  <Command>c:\a.bat</command>
                  <commandParams>p1 p2 p3</commandParams>
                  <SQLQry>select * from table</SQLQry>
                   <Run Status="No run"/>
                  <time start=" Will be filled by user" end=" Will be filled by user" />
               </step>
               <step id="55445" Order="2" >
                  <Description>HTML</Description>
                  <Expected>HTML</Expected>
                  <actual>Will be filled by user</actual>
                  <Command>c:\a.bat</command>
                  <commandParams>p1 p2 p3</commandParams>
                  <SQLQry>select * from table</SQLQry>
                   <Run Status="No run"/>
                  <time start=" Will be filled by user" end=" Will be filled by user" />
               </step>
            </steps>
            <InputTestParams>
               <param name="Band" value="2.5"/>
               <param name="BandWidth" value="35"/>
            </InputTestParams>
            <ResultParams >
               <param name="filled by user" Expected="by user" Actual="by user"/>
                More results Will be filled by user (include Run & test parameters)
            </ResultParams >
      </TestInExecLine >
```

Cycle is the one shown in the execution set (the last one)
Run name is a unique string from datetime parsing - created for each call for this web service and will be used to record the results with unique value.
Run version default is taken from the execution set version.
User is the login user to the Web Service (User_Name)
TestInExecLineID – the id of the line of the test in the execution. Each line is per test (can be duplicate due to parameters instances or due to double test in the execution).
ImportStatus – Success/Fail/Empty - Used when saving the results. Each test that was saved successfully is marked Success. Each test that was failed in save marked as Fail. Default is empty.
Test Params - Note that they will be added only if the instances of the test where added to the test before it added to the execution set (the test exist in the exe-set as instance). It will not be added if the test has parameters but not instances.
Results Params - Will be writen after the XML is created and will be used for 'Record_execution_Results' in order to define and simulate **new** parameters. Not yet implemented

Example(Visual Basic):

Open issues
Notes:
- Project & Version is not relevant since each execution in differnt versions has uniqe Id. Even if not, the execution was get different Id when branched (or save changes to previouse versions).
- When there is no test to write to the XML, the header is writen anyway.
- Result Parameters

2.20 get_filter_list_xml

Same as get_filter_list, result in XML and not string

2.21 Get_Item_Details

The system should return the Item Details report using web service without using QPackServer.dll

All hyperlinks from old format should be working with the new service.

2.22 Get_Item_Details_xml

2.23 Get_Object

Purpose

Return object details
See Input/Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Object

General Description

The web service will return object data in XML format
Field Description returns in HTML format
QPack item description could contain pictures/images so this web service will return the description also as 64 encode (fields Description64)

Input

Table - Columns and description for each column

Input:

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
ID	Long	Yes	Item ID

Output

If succeed:

```
<object id="3232" action="" ALManalyticsBIurl="http:\\BIserver\analytics\report">
    <field Name="Status">New</field>
 <field Name="Priority">Not Set</field>
 <field Name="Assigned To">Nimrod</field>
 <field Name="Version">1.0</field>
  <field Name="Name">Test reported test dfdd</field>
 <field Name="Due Date">30/12/2009</field>
<field Name="Category">Not set</field>
 <field Name="Description">
     <![CDATA[ <html>check item<html> ]]>
   </field>
 <field Name="Start Date">30/12/2009</field>
 <field Name="Object Type">Requirement</field>
 <field Name="ID">3</field>
 <field Name="Created By">Nimrod</field>
 <field Name="Created Date">30/12/2009 10:37:30</field>
 <field Name="Updated By">Nimrod</field>
 <field Name="Updated Date">23/02/2010 16:57:13</field>
 <field Name="Lock By">Ramia</field>
 <field Name="Effort Estimation">0</field>
 <field Name="OS">DD</field>
</object>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
```

```
res = objSoap.Get_Object ("pola", "1234",28)
```

Open issues

- Field Description64 is not available for now

2.24 Get_Object_Attachments

Purpose

Return list of item attachments
See Input /Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Object_Attachments

General Description

The web service will return list of item attachments.
If the attachment type is "File" then the service return the attachment as 64 encode

Input

Table - Columns and description for each column

Input:

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_Name	String	Yes	Project name
Version	String	Yes	The version view of the objects. Format shall be : [Major version]. [Minor version] (2.0 for ex.)
Object_ID	Long	Yes	Selected item id

Output

If succeed:

```
<object Id=”4343” AttachmentCount="4">
        <attachment ID=”4343”/>
                <fileName>”aaa.doc”</fileName>
                <AttachDescription>description of attachment</attachDescription>
                <AttachType>URL/FILE</attachType>
                <AttachContent64>fsdfdfg777f7sdfs88sgfdgtt</AttachContent64>
</attachment>
</object>
```

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Object_Attachments("pola",
"1234","Qpack","2.0",28)
```

Open issues

2.25 Get_Object_Audit_Trail

Purpose

Return list of item attachments
See Input/Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Object_Audit_Trail

General Description

The web service will return list of item audit trail (history)

The user can get more then one item audit trail and all results will return in one table

Input:

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Item_Key	String	Yes	Items keys (seperated with comma)
Version	Long	Yes	The version view of the objects (current project view)
From_Date	String	No	Get records from date.Format: DD/MM/YYYY

Output

If succeed:

```
<Table>
.
.
.
</Table>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
```

```
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Object_Audit_Trail("pola", "1234","REQ-123,TASK-88",2,"26/5/2008")
```

Open issues

2.26 Get_Object_Definition

Purpose

Gets data structure of objects per project
See Input/Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Object_Definition

General Description

Gets data structure of objects per project (Separate for each project since custom fields may be different for each project)
Includes the following T-plan API:

- Retrieve list of object types definitions
- Retrieve list of data types
- Retrieve list of attribute definitions for each type definition
- Object constraints will be validated during insert (for example – can't create DOC under REQUIREMENT – will be checked during insert, these constraints will not be transferred in this XML).
- If input project ID is empty – return all projects

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_ID	Long	No	Project ID

Output

If succeed:

```
<objects_per_project>
        <Project id=3232 name="project name">
        <object type = "requirement" icon="64 encode binary based array"/>
                <field Name="Object_name" Type ="String" />
                <field Name="Description" Type ="String" />
                <field Name="Description64" Type ="64 encode base" />
                …
                <Field name= "Browser" Type="String" />
                …
        </object>
    </project>
</objects_per_project >
```

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Object_Definition ("pola", "1234",2)
```

Open issues

- Object icon is not available for now

2.27 Get_object_discussion

This service gets the discussion of an item.

Parameter	Value
sUserName:	QPack user name
sPassword:	QPack usr password
sItemKey:	Any item key

2.28 Get_object_metadta

Name: Get_object_metadta

Input: **projecteId**, **ObjType**

Returns result in XML DATATABLE – will be used by lali

Query to run:

```
select item_data_name
as custom_label,column_type, item_data_name, i.item_order, is_custom, co
de, description, max_length, is_default, v.item_order
from item_data i left outer join sys_ref_values v
on REF_TABLE=TABLE_CODE
where project_id=PROJ_ID and obj_type='OBJ_TYPE'
union
select label_ctrl_name, control_type, gui_syntax, ctrl_screen_order, '0'
 , code, description,'100', is_default, v.item_order
from screen_controls s left outer join sys_ref_values v
on ref_table=table_code
where obj_type='OBJ_TYPE'
and user_control='ctlDetails'
and show_ctrl=1 and is_systemfield=1
order by i.item_order, v.item_order
```

2.29 Get_Object_Relations

Purpose

Return list of item relation

See Input/Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Object_Relations

General Description

The web service will return list of item relations (When item define as source or as target)

Only object relations - not include defects

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
ID	Long	Yes	Item ID

Output

If succeed:

```
<Object Id =”32322” RelationCount="0">
        <relation source =”32322” target =”545454” />
        <relation source =”3111” target =”32322” />
</object>
```

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Object_Relations("pola", "1234",28)
```

Open issues

2.30 Get_Projects

Purpose

Return list of projects and versions

See Input/ Output in http://[Server Name]/Qpack/qpackserv/qpackserv.asmx?op=Get_Projects

General Description
The web service will return list of projects and versions

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password

Output

If succeed:

```
<projects>
        <project id =”4343” name=”proj1” version =”1.0” type="Project"/>
        <project id =”4343” name=”proj1” version=”2.0” type="Project"/>
</projects>
```

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Get_Projects("pola", "1234")
```

Open issues

2.31 Get_Qpack_Xml

2.32 Get_Source_Report

2.33 Get_Unit_Execution_List

Purpose

This web service will return list of unit test execution sets from specific project

General Description

This web service will return list of unit test execution sets from specific project

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_Name	Long	Yes	Project Name
Version_View	String	Yes	The version of the selected project (2.0 for ex.)

Output

```
<execution_list>
    <execution Id=”1222” >
        <Name>Execution set 1</name>
        <Description>[CDATA]HTML description of execution[CDATA]/Description>
        <parentId>4433</parentId>
        <parentName>Task1<parentName>
    </execution>
    <execution Id=”3333” >
        <Name>Execution set 2</name>
        <Description>[CDATA]HTML description of execution[CDATA]/Description>
        <parentId>4444</parentId>
        <parentName>Task2<parentName>
    </execution>
</execution_list>
```

2.34 Get_User_Permissions_ByItemType

2.35 Get_user_projects

Works the same way as get_projects
Gets only active projects of the user
Used in Gantt

2.36 Get_User_Projects_Web

2.37 Get_Users

Purpose

Returns all users of selected project or all users list if no project specified

General Description

The web service will return a list of all authorized users in a specified project or all the active users in QPack

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Project_ID	String	Yes	Project ID or "" to retrieve all active users

Output

If succeed:

String array containing all user names (login names)

```
<?xml version="1.0" encoding="utf-8" ?>
rrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://orcanos.com/">
ring>zohar</string>
 <string>Nimrod</string>
 <string>Rami.Azulay</string>
rrayOfString>
```

If Fail:

```
<Error>
      <ErrorStatus>-1</ErrorStatus>
      <ErrorInfo>Error description</ErrorInfo>
      <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
Dim res As String, sUrl as string
sUrl ="http://serverName/Qpack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.GetUsers ("pola", "1234","Qpack")
```

Open issues

2.38 Get_Validation

Use related WS interface function **Get_Validation** that returns table for each **object type** selected
Example:

Field name	Possible Values	Is Default
Status	- Not Set -	1
Status	In Process	0
Status	Done	0
...	...	
Category	WS	0
Category	GUI	0
...	...	

Each value in the Exell imported - ***Data*** - , maped to QPack Item Field - ***Field*** - , should be checked against the table
Field -> **Field Name**
Data -> **Possible Values**

If ***Data*** is not found in **Possible Values** then validation check failed for the current line
If it is empty or null -> use default value from the table

- (Code taken from existing data migration programs and incorporated into the related WS new interface)
object types
DEFECT - **Done!**
T_CASE - **Done!**
REQ -
TASK -

2.39 Get_WebServ_Ver

2.40 GetActivityListByProjectVersionID

This service returns all items assigned to a specific user based on projects and versions.

2.41 GetALLObjTypes

This service creates a list of all items in QPack system based on project and version IDs

2.42 getFTPSettings

Purpose

This service returns the FTP settings of QPack server.

If **FTP_Read** param is defined in QPack.xml it will return both **FTP** and **FTP_Read** param's in this order
This will ensure backward compatibility with previous version returning **FTP** param only see related item

2.43 getProjectVersionList

This service returns all projects and versions based on given user name and password.
This service returns the result in XML format.

This function will accept User Name and Password

The function will check users name validation

The function will return the list of projects and versions that the user can access.

2.44 GetTree

This service returns QPack tree view structure based on project and version IDs in XML format.

2.45 Init

2.46 Insert

This service inserts new item into QPack pool or view

2.47 Insert_BYCmd

This service insert new item into QPack using all item parameters including optional and mandatory fields. Using this service it is possible to apply values other then default.

2.48 Lock_object

This service lock / unlock an object
Input : Object ID , Version ID

This service used by QPack application or word addin only

2.49 Record_Execution_Results

Purpose

This web service will record execution set results

General Description

This web service will get XML reflecting run details (execution set, instances & run results), and update QPack with this run.

Input

The XML this web service uses taken from Get_Execution_Run_Details, or Get_Execution_Run_Details_xml.

The XML can be updated (before calling the Web Service) with the following:

- **Tester** – the user executing the test – taken from QPack. User can override the value with valid user name. In test level
- **Time start/end** – User will input start and end time for execution and for each step
- **Run version** – change to actual version executed. In test level
- **Run Status** - Updated for each step. Values can be: Pass, Fail, Skipped. If other value is set (or no value) – status will remain "No Run".
  If some steps were changed for a specific test and some not – the step status will be remain "No Run" and test status will be set to "Not completed".
- **Actual** – Actual results (HTML format, for each step)
- **Resultparams** – Not in use

Output

1. QPack will return the following results:
   - **Pass** – save succeeded
   - **Fail** - If error occurred

2. Save results to the database

3. QPack will save the updated XML file in the working directory on QPack server, in folder (**c:\program files\qpack\automation**):

The Webservice will update the XML with the following:

- **ImportStatus** – Success/Fail/Empty - Used when saving the results. Each instance that was saved successfully is marked **Success**. Each test that was failed in save marked as **Fail (**do not mistake it with the test result – it's the web service saving to DB success criteria**)**. Default is empty (update the XML string and save as file at the end in the working directory after updating migration status).
- **ErrorDesc** - In case Import status is **Fail**.
- **Test Last Run Status** – Change according to the steps statuses & QPack methodology (see below)

Error management

If there was a failure – QPack import will stop. The **ImportStatus** and **ErrorDesc** will be updated accordingly. The description of the error message will be in XML format.

Test status

The status of the test will be derived from step results – based on existing QPack methodology:

- If one of the steps is Fail -> test run status is Fail
- Else If one of the steps is No-run -> run status is Not completed
- Else If one of the steps is Pass -> run status is Pass (few Pass & few Skip -> Pass)
- Else If all steps are skipped -> run status is Skip

(if Fail and after there is No-run -> Fail; there is no Fail-stop since it happens in QPack only when selecting to Quit.)

Additional issues

If import status is fail, user can fix the error and call the web service again with the same XML. QPack will not save results twice (managed by Run name. If run name exist in QPack database - review the XML and skip where importStatus ="success")

2.50 Send_Mail

The system should support 64bit OS, the new SendMail mechanism will be based on web service.
The service will support all previous functionalities such as:

- Email formatting
- Hyper link to Item details from EMail body
- Alert service send mail in report format
- Replace the mechnisim of the send mail from the QPack client.

Send mail – .net (don't forget to add reference to system.web)

1. example: http://www.vbdotnetheaven.com/UploadFile/prvn_131971/mailvb11172005000829AM/mailvb.aspx
2. Both QPack and ASP will call the new webservice
3. Consider use webservice and not ASP
4. Must be asynchronic- use async var as true when post ws
5. Compile as 64bit (Needs to work also on old OS such as Windows XP 32 bit)
6. Support password protected and Encrypted SMTP services
   Keeping **email 'From' clause free to be defined by the sender**

2.51 Subscribe_Item

This service can subscribe/unsubscribe users to specific item

Parameter	Value
Email:	Your Email
Item_Key:	For Example: DEFECT-4333
Subscribe:	1 for subscribe, 0 for unsubscribe

2.52 TestAccessConnection

This service return the parameter and if the function is succeed.

2.53 TestConnection

This service tests the support for 'XML' structure and system sanity result. This service is used for sanity tests after installation of QPack web services.

2.54 TestInsert

This service tests the insert operation for new item using mandatory fields.

2.55 TestUpdate

This service tests the update operation of item. This service allows to update only item Name and Description.

2.56 Unlock_object

2.57 Update

This service update QPack item using {username, password, object id and new values: name and description}

2.58 Update_Defect

2.59 Update_Object_By_ID

Purpose

Update object details in QPack.
See Input/Output in http://[Server Name]/QPack/qpackserv/qpackserv.asmx?op=Update_Object_By_ID

General Description

This web service will update item properties.
The web service will save up to 15 custom fields
changes will save to history
Not handle branches

The service will check the following:
1.Valid values for system and custom fields
2.user name + password validation

3.Item lock by other user
4.Status/version freeze

If Empty value then the service not update the specific field value.

Input

Field name	Type	Mandatory	Description
User_Name	String	Yes	Valid QPack user name
User_Password	String	Yes	Valid QPack user password
Object_ID	Long	Yes	Object ID
Object_Name	String	No	The item name
Description	String	No	Item description in plain text/HTML format
Release_Version	Long	No	Release value (major.minor.**RELEASE**.build)
Build_Version	Long	No	Build value (major.minor.release.**BUILD**)
Assigned_to	String	No	User name the object should be assigned to.
Category	String	No	Specific object category
Status	String	No	Valid object status
Priority	String	No	Valid object priority
Start_Date	String	No	The object starts date. **Format must be "DD/MM/YYYY"**
Effort_Estimation	Long	No	Object effort estimation
Due_date	String	No	The object due date. **Format must be "DD/MM/YYYY"**
CS1_Name	String	No	Object 1st custom field name.

Field name	Type	Mandatory	Description
CS1_value	String	No	Object 1st custom field value
Up to CS60			Up to 60 custom fields

Output

If succeed:

```
<Update>
    <Result>true</Result>
    <ErrorStatus>0</ErrorStatus>
</Update>
```

If Fail:

```
<Update>
    <Result>false</Result>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Update>
```

Example(Visual Basic):

```
dim objSoap as New MSSOAPLib30.SoapClient30
Dim res As String, sUrl as string
sUrl ="http://serverName/QPack/qpackserv/qpackserv.asmx"
objSoap.MSSoapInit (sUrl & "?wsdl")
res = objSoap.Update_object_By_ID ("pola", "1234",2, "New Object
Name","New description")
```

Open issues

- User permission edit specific item type
- Work flow is valid
- user group work flow permission

2.60 Update_Server_ConfigXML

Purpose

Update config XML files on server

General Description

The web service will receive xml file name, parameter name , parameter value
The web service will update the parameter in the specified xml file (QPack.xml or EmailFormat.xml) on QPack server
If parameter doesn't exist ot will create it with the specified value -

Input

Field name	Type	Mandatory	Description
Admin_Password	String	Yes	
File_Name	String	Yes	QPack.xml or EmailFormat.xml
Parameter_XPath	String	Yes	XPath expression describing the element/attribute to be updated
Parameter_Value	String	Yes	New value to set

2.61 UpdateCommitRevisionNumber

Purpose

Updates **object_source**.SCCommitRevisionNumber

where **object_source**.CommitFileName=input CommitFileName

General Description

The web service will update **object_source**.SCCommitRevisionNumber

Input

Field name	Type	Mandatory	Description

Field name	Type	Mandatory	Description
CommitFileName	String	Yes	
SCCommitRevisionNumber	String	Yes	

Output

If succeed: **OK**

If Fail:

```
<Error>
    <ErrorStatus>-1</ErrorStatus>
    <ErrorInfo>Error description</ErrorInfo>
    <ErrorTrace>Trace of error</ErrorTrace>
</Error>
```

Example(Visual Basic):

```
res = objSoap.UpdateCommitRevisionNumber("D:\RND\My Project\My Source File.code", "1.111")
```